Scrum Notes

Date: 10/12/2023
Time: 4pm
Location: Torg

Team Members Present:

Jae
Mohammad
Braden
Achintya

Agenda:

1. Update team membersTalk about next steps
2. Talk about any roadblocks
3.

Meeting Notes:

Jae:

- Tasks Completed:
    - PM1 Proposal
        - Abstract Section
        - Introduction Section
        - Related Work Section
    - Requirements Workshop
    - PM1 Lightning Slides
        - Problem to Solve Intro Slide
        - Solution Slide
- Tasks To Do:
    - PM2
        - Requirement Analysis Use Case

Mohammad:

- Tasks Completed:
    - Documented Scrum Meetings
    - Requirements Workshop
- Tasks To Do:
    - PM2
        - Project Scrum Meeting Deliverable
        - Requirement Analysis Use Case

Braden:

- Tasks Completed:

        - PM1 Lightning Slides
            - Related Work Slide
            - Example Slide
            - Software Engineering Process Slide
        - Requirements Workshop
- Tasks To Do:
    - PM2
        - Requirement Analysis Use Case

Achintya:

- Tasks Completed:
    - PM1 Proposal
        - SWE Process Section
    - Requirements Workshop
- Tasks To Do:
    - PM2
        - Requirement Analysis Use Case

Use Cases

Use Case 1: Researching Programming Definitions

Preconditions:

The user is in the IDE.

The user has a programming-related question.

Main Flow:

The user activates the search engine from within the IDE.

The user types in a query related to the programming term or concept.

The search engine retrieves relevant definitions and explanations.

Results are displayed in a user-friendly manner within the IDE.

Alternative Flows:

If no relevant results are found, the system suggests alternative search terms.

Use Case 2: Help with Code Solutions

Preconditions:

The user is working on a specific code problem in the IDE.

The user needs help with code implementation.

Main Flow:

The user activates the search engine.

The user describes the coding problem they are facing.

The engine suggests relevant code solutions or snippets.

The user reviews and implements the proposed solution.

Alternative Flows:

The engine prompts for more details if the user's query is ambiguous.

Use Case 3: AI-Assisted Code Debugging

Preconditions:

The user has a piece of code that is producing an error.

The user is unsure about the cause of the error.

Main Flow:

The user highlights the problematic code and activates the search engine.

The engine analyzes the code and identifies potential issues.

Suggestions for fixing the error are presented to the user within the IDE.

Alternative Flows:
It will tell you where it thinks it is going wrong

Use Case 4: Code Implementation Assistance

Preconditions:

The user is trying to implement a specific functionality but doesn't know the best approach.

Main Flow:

The user describes the desired functionality to the search engine.

The engine provides potential code implementations or algorithms that can achieve the desired functionality.

User reviews and integrates the suggested approach into their project.

Alternative flows:

It will reference similar solutions and provide them to the user

Use Case 5: Context-Based Code Assistance

Preconditions:

The user is actively writing code in the IDE.

Users might be unsure about the next steps in the code.

Main Flow:

The search engine continuously monitors the user's coding activity to detect pausing for an extended time after typing a partial statement.

When a pause is detected, the engine proactively suggests potential code completions or next steps based on the current code context.

The user reviews the suggestions and can choose to accept, modify, or ignore them.

Alternative Flows:

If the user finds the suggestions distracting, they can adjust the sensitivity or turn off this feature.

Postconditions:

The user has been provided with relevant code suggestions to aid in their coding process.

Requirements Analysis

1. Provide an example of five hypothetical non-functional requirements for this system. Be sure to include the specific type of requirement discussed in class, with each requirement coming from a unique category.
    1. Performance

Category: Performance requirements ensure that the system meets specific response time criteria.
Requirements: The search engine must provide code suggestions within 2 seconds of a user's query input, ensuring a responsive user experience.

2. Security
   Category: Security requirements focus on protecting data and system integrity.
   Requirements: All code snippets and project data transferred to and from the search engine must be encrypted using industry-standard encryption protocols to ensure data security and confidentiality.
3. Usability
   Category: Usability requirements address how the system is presented and used by the end-users.
   Requirements: The search engine must provide a user-friendly interface with customizable code suggestion settings, allowing developers to tailor the engine's behavior to their coding style and preferences.
4. Scalability
   Category: Scalability requirements focus on the system's ability to handle increased loads as the user base grows.
   Requirements: The search engine should be able to scale horizontally to accommodate a growing user base. It should support at least a 50% increase in concurrent users within six months of deployment.
5. Compliance
   Category: The search engine must include functionality to automatically check and alert developers to potential licensing and copyright violations when suggesting code snippets, promoting compliance with open-source licenses and copyright laws.
   Requirements: Compliance requirements ensure adherence to legal and regulatory standards.

2. Provide an example of five hypothetical functional requirements for this system.
   1. Code Documentation
      Category: Documentation Assistance requirements pertain to features that assist developers in creating and accessing code documentation.
      Requirements: The search engine should automatically generate documentation for code snippets, including function descriptions, parameter details, and usage examples. This documentation should be accessible to developers as a pop-up or sidebar within the IDE.
   2. Search Functionality
      Category: Search and Query requirements specify how the system handles user queries and code context.
      Requirements: The search engine should be capable of analyzing the context in which a user's code is written. It should consider variables, function names, and project structure to provide more contextually relevant code suggestions.
   3. Version Control

Category: Integration requirements relate to how the system interacts with external tools and systems.
Requirements: The search engine should integrate with Git version control systems, allowing developers to view code history, track changes, and access code from different branches. Users should be able to commit, push, and pull code changes directly from the IDE.

4. Collaboration
   Category: Collaboration requirements focus on features that support teamwork and communication.
   Requirements: The search engine should enable real-time code collaboration among team members. Users should be able to share code snippets, work together on the same code file, and provide comments and annotations within the IDE.
5. Recommended Algorithm
   Category: Recommendation requirements involve defining the system's ability to provide intelligent, data-driven suggestions.
   Requirements: The search engine should employ machine learning algorithms to continually improve code suggestion accuracy. It should learn from user behavior, such as code selections and refusals, and refine its recommendations based on this feedback.

3. Think of a specific task required to complete each of the functional requirements and non-functional requirements mentioned above (10 total). Estimate the amount of effort needed to complete this task using function points (i.e., using the values here). Briefly explain your answer.

1. Performance
   Estimating the effort required to meet the performance requirement of providing code suggestions within 2 seconds of a user's query input involves considering the complexity of the task and the potential impact on the system's performance, which could take around 2 to 6 months.. This particular performance requirement may have a moderate level of complexity because it involves optimizing the search and recommendation algorithm to deliver timely responses without compromising the quality of code suggestions.
2. Security
   To estimate the effort needed for the specific task of implementing data encryption for all code snippets and project data transferred to and from the search engine, we can consider the complexity of the task and the potential impact on the system's security, which could take around 1 to 3 months. This task involves implementing encryption for data in transit, which is a fundamental security practice.
3. Usability
   To estimate the effort needed for a specific task related to the usability requirement, such as providing a user-friendly interface with customizable code suggestion settings, we can consider the complexity and impact of the task on

the overall usability of the system, which could take around 2 to 6 months to implement.

4. Scalability

   To estimate the effort needed for the specific task of making the search engine scalable to support a 50% increase in concurrent users within 3 to 6 months of deployment, we should consider the complexity of the task and its potential impact on the system's scalability.

5. Compliance

   To estimate the effort needed for the specific task of implementing functionality to automatically check and alert developers to potential licensing and copyright violations when suggesting code snippets, we can consider the complexity of the task and the impact on the system's compliance with legal and regulatory standards, which would take around 3 to 6 months to do. This task involves integrating a feature into the code search engine that scans code snippets and project data for potential licensing and copyright issues.

6. Code Documentation

   To estimate the effort needed to implement the specific task of automatically generating code documentation for code snippets, including function descriptions, parameter details, and usage examples, we need to consider the complexity and scope of the task. This whole process could around 2 to 6 months.

7. Search Functionality

   The effort required for this task can vary widely, depending on the current state of the search engine and the complexity of the existing algorithms. A rough estimate could be in the range of 3 to 6 months. The task involves developing and implementing algorithms for contextual code analysis. This includes considering variables, function names, and project structure to understand the context of the code being written.

8. Version Control

   Effort estimation for these tasks can vary based on factors like the complexity of the existing system, the level of integration required, and the experience of the development team. The integration with Git version control systems could require several months, typically ranging from 2 to 6 months or more, depending on the depth of integration and the features required. It's essential to allocate time for testing and user feedback to ensure that the integration is robust and meets the needs of the developers using the IDE.

9. Collaboration

   The effort required for implementing real-time code collaboration features can be moderate to high, and it may take approximately 3 to 6 months, depending on the specific functionality and integration complexity. The estimated effort is a rough approximation and can vary based on the complexity of the features, the size of the development team, and the level of expertise in real-time collaboration development. Additionally, ongoing maintenance and updates may be required to ensure a seamless collaborative coding experience for users.

10. Recommended Algorithm

The specific task required to complete the requirement of employing machine learning algorithms to continually improve code suggestion accuracy involves implementing and maintaining a recommendation algorithm that can adapt based on user behavior, which could take 4 to 12 months due to the complexity. This task involves designing and developing a machine learning recommendation algorithm that can analyze user behavior, including code selections and refusals. This requires expertise in machine learning, data science, and software development.

4. Write three user stories from the perspective of at least two different actors. Provide the acceptance criteria for these stories.

1. User Story 1 (actor: developer): Code Suggestions

While I am coding a project in my IDE, I want the search engine to provide code suggestions for a method as I am working on it because I don't know the most efficient way to implement it.

Acceptance Criteria:

1. When I start typing in my method that I am implementing, the search engine will provide a suggestion within 2 seconds.
2. The suggestion should be relevant to the entire project that I am working on, such as variable names, function names, classes, and coding language.
3. I should be able to click a button in order for the suggestion code to be implemented.
4. If I don't like the suggestion, I should be able to decline the suggestion.
5. The search engine should then test why that suggestion didn't work correctly for the project and learn from it.

2. User Story 2 (actor: developer): Code Documentation

While I am coding in my IDE, I want to accept the suggested code provided to me by the search engine. The search engine should also provide documentation as why and how the code works so I can understand it more clearly.

Acceptance Criteria:

1. When I hover over suggested code, there should be a pop up describing in detail why the code works with the project as a whole.
2. The documentation should be concise and informative in order to help me understand the purpose and usage of the code.
3. The documentation should be up-to-date and relevant to the code suggestion.

3. User Story 3 (actor: administrator): User Behavior Analysis

I want to monitor how users have been using the search engine and I want to know if the code suggestions have been helpful to the users.

Acceptance Criteria:

1. I should have access to the surveys that people have filled out in order to better understand how the search engine has been functioning.
2. The system should track which code suggestions have been accepted and declined.
3. I should be able to export the user data for further analysis.
4. The suggestion algorithm should continuously learning from user behavior how to adjust and improve code suggestions.
5. I should be able to customize the algorithm's learning parameters.

5. Provide two examples of risk that could potentially impact this project. Explain how you would mitigate these risks if you were implementing your project as a software system.

1. Data Privacy and Security

   There are a lot of things that are needed to be done in order to ensure customers information is secured. All data needs to be encrypted using industry-standard encryption. Anonymizing code snippets and data, so that the search engine doesn't store or process sensitive information directly. In order to stay on top of any security issues, there would have to be a cybersecurity team that continuously checks on all leaks of information that could have happened.

2. Quality of Code Suggestions

   To ensure that our search engine suggests the best possible solutions to customers' coding projects, there needs to be rigorous testing and quality assurance process that includes test cases for code suggestions. Continuously evaluate and refine the suggestion algorithms based on feedback and test results. There also needs to be implementation to some sort of machine learning tool. regularly fine-tune the models with a feedback loop that learns from user behavior and adjusts the recommendations accordingly. When providing code suggestions, include explanations or reasons for the suggestions, helping developers understand why a specific piece of code is recommended.

6. Describe which process your team would use for requirements elicitation from clients or customers, and explain why.

To receive feedback from customers about how our search engine is functioning, there will be surveys for the customers to fill out after the user has used our software for 2 hours. After the first 2 hours, the survey will be asked again after every 10 hours of use. The reason for the first 2 hours is so that the user can get comfortable with the search engine and understand how it works. 2 hours is good amount of time to figure out the pros and cons of the software our group believes. The reason for every 10 hours after the initial 2 hours is so that the survey doesn't start to annoy the customer. After awhile of the customer using the search engine, they will start to recognize more specific suggestions for how to improve the software. We will then have a team look at all the surveys and determine the best ways to fix the problems.